Automated Synthesis of a Finite Complexity Ordering for Saturation

Yannick Chevalier [1] Mounira Kourjieh [2]

[1] IRIT, Université de Toulouse, France
[2] LORIA-CNRS, Nancy, France

Abstract. We present in this paper a new procedure to saturate a set of clauses with respect to a well-founded ordering on ground atoms such that $A \prec B$ implies $VarA \subseteq \mathrm{Var}(B)$ for every atoms A and B. This condition is satisfied by any atom ordering compatible with a lexicographic, recursive, or multiset path ordering on terms. Our saturation procedure is based on a priori ordered resolution and its main novelty is the on-the-fly construction of a finite complexity atom ordering. In contrast with the usual redundancy, we give a new redundancy notion and we prove that during the saturation a non-redundant inference by a priori ordered resolution is also an inference by a posteriori ordered resolution. We also prove that if a set S of clauses is saturated with respect to an atom ordering as described above then the problem of whether a clause C is entailed from S is decidable.

1 Introduction

Resolution is an inference rule introduced by Robinson [14] for theorem proving in first-order logic. It consists in saturating a theory presented by a finite set of disjunctions, called clauses, with all its consequences. Since the seminal work of Robinson, lot of efforts have been devoted to finding strategies that limit the possible inferences but still are complete for refutation. The correctness of resolution implies the correctness of these strategies. Among these we note selected resolution [3] and ordered resolution [2] which are correct and refutationally complete. The later being a special case of [7]. Later, it was proved in [4] that if a set S of clauses is saturated by ordered resolution (with some additional hypotheses discussed in this paper) then deciding whether a clause C is a consequence of S is decidable. We present in this paper a weakening of the hypotheses assumed in [4] that also enjoys this decidability property. In [4], it is proved that saturated sets of clauses are order local, and thus if each atom has a finite number of smaller atoms then the ground entailment problem is decidable. Orders having this property are said to be of *finite complexity*.

We present in this paper a variant of the standard saturation procedure that builds during saturation an *atom rewriting system*. This rewriting system defines a partial ordering on ground atoms that has a finite complexity. Under our redundancy notion, we prove that the saturation of a set S of clauses implies its locality (as in [4]) with respect to the ordering based on the *atom rewriting*

system. As a consequence, if a set S of clauses is saturated with respect to an atom ordering as described above then the problem of whether a clause C is entailed from S is decidable. Finally we prove that the conditions imposed on the atom ordering are satisfied by all atom ordering compatible with a well-founded, monotone, and subterm term ordering, *i.e.*, most of the standard term orderings.

Outline of this paper. In Section 2, we present the basic notions that we will use later in the paper, in Section 3 we present some of the decidability results for the *ground entailment problem* given in the literature, in Section 4 we present our definitions of *atom rewriting system*, *locality* and *redundancy*, in Section 5 we give our *saturation procedure*, in Section 6 we give our decidability result, and in Section 7 we show how our result extends the results presented in Section 3.

2 Formal setting

2.1 Basic notions

Syntax. We assume that we have an infinite set of variables $\mathcal{X}$, an infinite set of constant symbols $\mathcal{C}$, a set of predicate symbols $\mathcal{P}$ and a set of function symbols $\mathcal{F}$. We associate the function *arity* to function symbols and predicate symbols, $arity : \mathcal{F}\cup\mathcal{P} \to \mathbb{N}$. The arity of a function symbol (respectively predicate symbol) indicates the number of arguments that the function symbol (respectively the predicate symbol) expects. We define the set of *terms* $\mathcal{T}(\mathcal{F},\mathcal{X})$ as follows: $\mathcal{X},\mathcal{C} \subseteq \mathcal{T}(\mathcal{F},\mathcal{X})$, and for each function symbol $f \in \mathcal{F}$ with arity $n \geq 0$, for each terms $t_1,\ldots,t_n \in \mathcal{T}(\mathcal{F},\mathcal{X})$, we have $f(t_1,\ldots,t_n) \in \mathcal{T}(\mathcal{F},\mathcal{X})$. We denote by $\mathrm{Var}(t)$ the set of variables occurring in the term t, and a term t is said to be *ground* if $\mathrm{Var}(t) = \emptyset$. We define subterms of a term t, denoted $Sub(t)$, as follows: if t is a constant or a variable then $Sub(t) = \{t\}$, if $t = f(t_1,\ldots,t_n)$ then $Sub(t) = \{t\} \cup \bigcup_{i\in\{1,\ldots,n\}} Sub(t_i)$. We denote by $t[s]$ a term t containing s as subterm. We define *atoms* as follows: if I is a predicate symbol in $\mathcal{P}$ with arity $n \geq 0$ and $t_1,\ldots,t_n$ are terms in $\mathcal{T}(\mathcal{F},\mathcal{X})$ then $I(t_1,\ldots,t_n)$ is an atom. A *literal* L is either A or $\neg A$ where A is an atom and $\neg$ denotes the *negation*. The literal L is a positive literal in the first case, and a negative literal in the second. We denote by $\mathrm{Var}(A)$ the set of variables occurring in the atom A and an atom A is said to be *ground* if $\mathrm{Var}(A) = \emptyset$.

A *clause (or full clause)* is defined by a set of literals $\{\neg A_1,\ldots,\neg A_m, B_1,\ldots,B_n\}$. It may also be viewed as a formula of the form $\Gamma \to \Delta$ where $\Gamma = \{A_1,\ldots,A_m\}$ and $\Delta = \{B_1,\ldots,B_n\}$; Γ represents the *antecedent of the clause* and Δ its *succedent*. We denote $Atoms(C)$ the set of atoms occurring in the clause C. A clause is *ground* if all its atoms are ground. A clause $\Gamma \to \Delta$ is *Horn* when Δ is a singleton or empty, and *unit* when it has only one literal. A clause $\Gamma \to \Delta$ is *positive* when it has only a succedent, *i.e.* $\Gamma = \emptyset$ and is *negative* when it has only an antecedent, *i.e.* $\Delta = \emptyset$. We write Γ_1,Γ_2 to indicate the union of the two sets Γ_1 and Γ_2 and usually omit braces. For example, we write Γ, A or A, Γ for the union of $\{A\}$ and Γ or write

$A_1, \ldots, A_m \to B_1, \ldots, B_n$ for $\{A_1, \ldots, A_m\} \to \{B_1, \ldots, B_n\}$. We also make more simplifications, for example we write A to denote the positive unit clause $\emptyset \to A$, and $\neg A$ to denote the negative unit clause $A \to \emptyset$. Let C be a clause, we denote by $\neg C$ the set of unit clauses $\neg L$ with L a literal in C; For example, $\neg C = \{A_1, \ldots, A_m, \neg B_1, \ldots, \neg B_n\}$ when $C = A_1, \ldots, A_m \to B_1, \ldots, B_n$. We say that a term t occurs in an atom A if A is of the form $I(\ldots, u, \ldots)$ with t a subterm of u and t occurs in a clause if it occurs in an atom of the clause.

Substitutions and unifiers. A substitution σ is a partial function from variables $\mathcal{X}$ to terms $\mathcal{T}(\mathcal{F}, \mathcal{X})$ such that $Supp(\sigma) = \{x | \sigma(x) \neq x\}$ is a finite set and $Supp(\sigma) \cap Var(Ran(\sigma)) = \emptyset$ with $Ran(\sigma) = \{\sigma(x) | x \in Supp(\sigma)\}$. We denote by $Var(\sigma)$ the set $Var(Ran(\sigma))$. A substitution σ with $Supp(\sigma) = \emptyset$ is called the *empty substitution* or the *identity substitution*. A substitution σ is said to be *ground* if $Var(\sigma) = \emptyset$, that is $Ran(\sigma)$ is a set of ground terms. A *renaming* ρ is an injective substitution such that $Ran(\rho) \subseteq \mathcal{X}$. A substitution σ is *more general* than a substitution τ, and we note $\sigma \leq \tau$, if there exists a substitution θ such that $\sigma\theta = \tau$. Equivalent substitutions, *i.e.* substitutions σ and τ such that $\sigma \leq \tau$ and $\tau \leq \sigma$ are said to be equal up to renaming since in that case the substitution θ is a bijective mapping from variables to variables. If M is an expression (i.e. a term, an atom, a clause, or a set of such objects) and σ is a substitution, then $M\sigma$ is obtained by applying σ to M as usually defined; We say that $M\sigma$ is an *instance* of M and if $M\sigma$ is ground we say that σ is *grounding* M.

A substitution σ is said to be a *unifier* of two elements (*i.e.* terms or atoms) e_1, e_2 if $e_1\sigma = e_2\sigma$. We denote $\mathrm{Unif}(e_1, e_2)$ the set of unifiers of the two elements e_1 and e_2. It is well-known that whenever the set $\mathrm{Unif}(e_1, e_2)$ is not empty it has a unique minimal element up to renaming. This minimal element is called the *most general unifier* of e_1 and e_2, and is denoted $\mathrm{mgu}(e_1, e_2)$.

Orderings. A (strict) ordering $\succ$ on a set of elements E is a transitive and irreflexive binary relation on E. The ordering $\succ$ is said to be:

- *total* if for any two different elements $e, e' \in E$, we have either $e \succ e'$ or $e' \succ e$;
- *well-founded* if there is no infinite descending chain $e \succ e_1 \succ \ldots$ for any element e in E;
- *monotone* if $e \succ e'$ then $e\sigma \succ e'\sigma$ for any elements e, e' in E and any substitution σ.

Any ordering $\succ$ on a set of elements E can be extended to an ordering $\succ^{set}$ on finite sets over E as follows: if η_1 and η_2 are two finite sets over E, we have $\eta_1 \succ^{set} \eta_2$ if (i) $\eta_1 \neq \eta_2$ and (ii) for every $e \in \eta_2 \setminus \eta_1$ then there is $e' \in \eta_1 \setminus \eta_2$ such that $e' \succ e$. Given a set η_1, a smaller set η_2 is obtained by replacing an element in η_1 by a (possibly empty) finite set of strictly smaller elements. We call an element e *maximal* (respectively *strictly maximal*) with respect to a set η of elements if for any element $e' \in \eta$ we have $e' \not\succ e$ (respectively $e' \not\succeq e$). If

the ordering $\succ$ is total (respectively well-founded and monotone), so is its set extension.

We denote by an *atom ordering* $\succ_a$ (respectively *term ordering* $\succ_t$) any arbitrary ordering on atoms (respectively on terms). We extend an *atom ordering* $\succ_a$ to a *clause ordering* as follows: we identify clauses with the sets of their respective atoms, and we order the clauses with respect to the sets of their respective atoms using the ordering ${\succ_a}^{set}$. For example, the clauses $A_1, A_2 \to B$ and $A_1 \to B$ are identified respectively with the following sets of atoms $\{A_1, A_2, B\}$ and $\{A_1, B\}$; The second set is strictly smaller than the first one with respect to the ordering ${\succ_a}^{set}$, and hence the second clause is strictly smaller than the first one.

In the remainder of this paper, we assume that the atom ordering $\succ_a$ is monotone, well-founded, and is such that $A \prec_a B$ implies $\mathrm{Var}(A) \subseteq \mathrm{Var}(B)$ for every atoms A and B.

2.2 Resolution

The resolution is an inference rule introduced by Robinson [14]; It is one of the most successful methods for automated proof search in first-order logic. We say that a set S of clauses is *unsatisfiable* if there is no *Herbrand interpretation* satisfying it, and *satisfiable* otherwise. Given a set S of clauses and a ground clause C, $S \models C$ means that C is true in every Herbrand model of S; It is easy to see that $S \models C$ iff $S \cup \neg C$ is unsatisfiable. A *proof by refutation* of $S \models C$ consists in proving that $S \cup \neg C$ is unsatisfiable. The resolution has been proved in [14] to be correct and complete for refutation. The correctness of the resolution means that the empty clause (*i.e.* a contradiction) can not be derived from any satisfiable set of clauses, and the completeness means that the empty clause can be derived from any unsatisfiable set of clauses.

The *resolution* is described by the two inference rules given in Fig. 1. The clause $(\Gamma, \Gamma' \to \Delta, \Delta')\alpha$ of the resolution rule is called the *resolvent* of the premises ($\Gamma \to \Delta, A$ and $A', \Gamma' \to \Delta'$) or the *conclusion* of the inference, and the atom $A\alpha$ is called the *resolved* atom. In the factoring rule, the clause $(\Gamma \to \Delta, A)\alpha$ is called the *factor* of the premise $(\Gamma \to \Delta, A, A')$ or the *conclusion* of the inference, and the atom $A\alpha$ is called the *factored* atom.

$$\frac{\Gamma \to \Delta, A \qquad A', \Gamma' \to \Delta'}{(\Gamma, \Gamma' \to \Delta, \Delta')\alpha}$$

where $\alpha = \mathrm{mgu}(A, A')$.

(a) *Resolution rule.*

$$\frac{\Gamma \to \Delta, A, A'}{(\Gamma \to \Delta, A)\alpha}$$

where $\alpha = \mathrm{mgu}(A, A')$.

(b) *Factoring rule.*

Fig. 1: Standard resolution and factoring rules

Ordered resolution. Since the seminal work of Robinson [14] lot of efforts have been devoted to finding strategies that limit the possible inferences but still are complete for refutation and correct; The correctness of these strategies is obtained from the correctness of the resolution. Among these strategies, there is the ordered resolution [1] which is used in this paper and will be presented in this paragraph.

The ordered resolution makes use of an atom ordering $\succ_a$ and is described by two inference rules: *ordered factoring rule* and *ordered resolution rule*. We distinguish two types of *ordered resolution*: the *posteriori ordered resolution* and the *priori ordered resolution*.

$$\frac{\Gamma \to \Delta, A \qquad A', \Gamma' \to \Delta'}{(\Gamma, \Gamma' \to \Delta, \Delta')\alpha}$$

where $\alpha = \text{mgu}(A, A')$, $A\alpha$ is strictly maximal with respect to $\Gamma\alpha$, $\Delta\alpha$ for $\succ_a$, and $A\alpha$ is maximal with respect to $\Gamma'\alpha$, $\Delta'\alpha$ for $\succ_a$.

(a) Posteriori ordered resolution rule.

$$\frac{\Gamma \to \Delta, A, A'}{(\Gamma \to \Delta, A)\alpha}$$

where $\alpha = \text{mgu}(A, A')$, $A\alpha$ is strictly maximal with respect to $\Gamma\alpha$ for $\succ_a$, and maximal with respect to $\Delta\alpha$ for $\succ_a$.

(b) *Posteriori ordered factoring rule.*

$$\frac{\Gamma \to \Delta, A \qquad A', \Gamma' \to \Delta'}{(\Gamma, \Gamma' \to \Delta, \Delta')\alpha}$$

where $\alpha = \text{mgu}(A, A')$, A is maximal with respect to Γ, Δ for $\succ_a$, and A' is maximal with respect to Γ', Δ' for $\succ_a$.

(c) *Priori ordered resolution rule.*

$$\frac{\Gamma \to \Delta, A, A'}{(\Gamma \to \Delta, A)\alpha}$$

where $\alpha = \text{mgu}(A, A')$, A is maximal with respect to Γ and Δ for $\succ_a$.

(d) *Priori ordered factoring rule.*

Fig. 2: Posteriori and priori ordered resolution and factoring rules.

Remarks.

1. The *posteriori* ordered resolution is actually the *ordered* resolution introduced in [1] and the *priori* ordered resolution is related to the *selected resolution* which is widely studied in the literature [9].
2. We remark that the two types of ordered resolution coincide on ground clauses, but not on non-ground clauses: let us consider the following two clauses $C = I(b, y) \to I(x, y)$ and $D = I(a, b) \to \emptyset$ and the ordering: $I(a, b) \prec_a I(b, b)$. We have $\sigma = \{x \mapsto a, y \mapsto b\}$ is the most general unifier of $I(x, y)$ and $I(a, b)$. We remark that $I(a, b)$ is maximal with respect to $\emptyset$, $I(x, y)$ and $I(b, y)$ are not comparable and hence $I(x, y)$ is maximal with respect to $I(b, y)$. This implies that the *priori* ordered resolution inference rule can be applied to the clauses C and D but not the *posteriori* ordered inference rule since $I(x, y)\sigma = I(a, b) \prec_a I(b, b)$. We remark that in the case of monotone atom ordering as we consider in this paper, the *posteriori ordered resolution* is included in the *priori ordered resolution*.

In spite of this difference between priori and posteriori ordered resolution, we introduce a redundancy notion such that every non-redundant priori ordered resolution inference is a posteriori ordered resolution inference (see Lemma 4).

It is well-known that the *posteriori* ordered resolution and the *priori* ordered resolution are correct and complete [14,1].

Ground entailment problem. The *ground entailment problem* studied in this paper is defined as follow:

> Given a set S of clauses, the *ground entailment problem for S* is defined as follows:
>
> **Input:** a ground clause C.
> **Output:** "entailed" if and only if $S \models C$.

3 Decidable fragments of first order logic

It is known that the ground entailment problem for Horn clauses and full clauses sets is undecidable in general. Here, we mention decidability results for some fragments.

3.1 McAllester's result

In [11], D. McAllester was interested by Horn clauses. He first defined the subterm locality as follows: a set S of Horn clauses is *subterm local* if for every ground Horn clause C, we have $S \models C$ if and only if C is entailed from a set of ground instances of clauses in S in which each term is a subterm of a ground term in S or in C. It is proved in [11] that if a set S of Horn clauses is finite and subterm local then its ground entailment problem is decidable.

3.2 Basin and Ganzinger results

In [4], D. Basin and H. Ganzinger generalized McAllester's result by allowing monotone, total, well-founded ordering over terms, and full (not Horn) clauses. To this end, they introduced several notions and we recall next some of them. A set of clauses S is said to be *order local* with respect to a term ordering $\succ_t$ if for every ground clause C, we have $S \models C$ if and only if C is entailed from a set of ground instances of clauses in S in which each term is smaller than or equal to some term in C. It is proved in [4] that if a set S of clauses is saturated up to redundancy by posteriori ordered resolution for a total, monotone, well-founded atom ordering then S is order local.

A term ordering $\succ_t$ is said to be of complexity f, g, whenever for each clause of size n (the size of a term is the number of nodes in its tree representation and the size of a clause is the sum of sizes of its terms) there exists $O(f(n))$ terms that are smaller than or equal to a term in the clause, and that they may

be enumerated in time $g(n)$. D. Basin and H. Ganzinger obtained the following decidability results:

1. If S is a set of (full) clauses that is order local with respect to a term ordering $\succ_t$ of complexity f, g then the ground entailment problem for S is decidable.
2. If S is a set of (full) clauses saturated up to redundancy by posteriori ordered resolution with respect to a complete well-founded atom ordering, and if, for each clause in S, each of its maximal atoms contains all the variables of the clause, then the ground entailment problem for S is decidable.
3. However, if the restriction on the variables in maximal atoms (the condition in the previous point) is removed, the ground entailment problem becomes undecidable in general.

We prove in this paper that it is possible to partially remove the condition on variables mentioned above while keeping the decidability of the ground entailment problem. More precisely: we prove in Theorem 1 the decidability of the ground entailment problem for S when S is a finite saturated set of clauses with respect to an atom ordering which is well-founded, monotone and such that $A \prec_a B$ implies $\mathrm{Var}(A) \subseteq \mathrm{Var}(B)$ for every atoms A and B.

The next three sections are devoted to this result.

4 Locality and redundancy

We introduce an atom rewriting system to model a new ordering relation between atoms. Our goal is to restrict the atom ordering $\prec_a$ to an ordering $\prec_{\mathcal{R}}$ such that each atom has only a finite number of predecessors.

Definition 1. *(Rewriting system on atoms.) Given an atom ordering $\succ_a$, we define a* rewriting system $\mathcal{R}$ on atoms *as a set of rules $L \to R$ where L and R are two atoms with $L \succeq_a R$.*

We give next some definitions that we use later in this section.

Definition 2. *Let A and B be two atoms, C be a clause and $\mathcal{R}$ a rewriting system on atoms. We have:*

- $A \downarrow_{\mathcal{R}} = \{B$ *such that* $A \to^*_{\mathcal{R}} B\}$, *i.e.* $A \downarrow_{\mathcal{R}}$ *denotes the set of atoms reachable from A when applying rules in $\mathcal{R}$.*
- $C \downarrow_{\mathcal{R}} = \{A \downarrow_{\mathcal{R}}$ *such that A is an atom in C* $\}$.
- $A \downarrow_{\mathcal{R}^-} = A \downarrow_{\mathcal{R}} \setminus \{A\}$.
- $C \downarrow_{\mathcal{R}^-} = \{A \downarrow_{\mathcal{R}^-}$ *such that A is an atom in C* $\}$.
- $A \prec_{\mathcal{R}} B$ *whenever* $A \in B \downarrow_{\mathcal{R}^-}$.

Lemma 1. *Let A and B be two distinct atoms. We have that $A \to_{\mathcal{R}} B$ implies $A \succ_a B$; And $A \prec_{\mathcal{R}} B$ implies $A \prec_a B$ and* $\mathrm{Var}(A) \subseteq \mathrm{Var}(B)$.

Proof. Let A and B be two distinct atoms such that $A \rightarrow_{\mathcal{R}} B$, then there exists a rule $L \rightarrow R \in \mathcal{R}$, a substitution σ such that $A = L\sigma$ and $B = R\sigma$. By definition of $\mathcal{R}$, we have $L \succeq_a R$ and then, by monotonicity of $\succ_a$, $L\sigma = A \succeq_a R\sigma = B$. Since A and B are different, we conclude that $A \succ_a B$. Now we assume that $A \prec_{\mathcal{R}} B$, this implies that $A \in B \downarrow_{\mathcal{R}^-}$, and hence $B \rightarrow^*_{\mathcal{R}} A$. Since $A \neq B$ we then have $B \succ_a A$. Since $A \prec_a B$ implies $\mathrm{Var}(A) \subseteq \mathrm{Var}(B)$ (by hypothesis on the ordering $\succ_a$), we then have $A \prec_{\mathcal{R}} B$ implies $\mathrm{Var}(A) \subseteq \mathrm{Var}(B)$.

Lemma 2. *Let $\mathcal{R}$ be a finite rewriting system on atoms. If A is a ground atom then the set $A \downarrow_{\mathcal{R}}$ is finite.*

Proof. Let A be a ground atom. By Lemma 1, we have $A \downarrow_{\mathcal{R}}$ is a set of ground atoms. Consider that graph $G = (A \downarrow_{\mathcal{R}}, V)$ where $(D, D') \in V$ if and only if $D \neq D'$ and $D \rightarrow_{\mathcal{R}} D'$. By Definition 2, $(D, D') \in V$ implies $D \succ_{\mathcal{R}} D'$. Thus G is acyclic. Since $\mathcal{R}$ is finite and $\mathrm{Var}(R) \subseteq \mathrm{Var}(L)$ for every rule $L \rightarrow R \in \mathcal{R}$, each node has a finite number of direct successor nodes. By $K\ddot{o}nig's$ lemma, if the graph G is infinite it has an infinite path. The atoms on this infinite path form an infinite strictly decreasing sequence of atoms $A \succ_a A_1 \succ_a A_2 \succ_a \ldots$ which contradicts the well-foundness of $\succ_a$. We then conclude that the graph G is finite, and hence is the set $A \downarrow_{\mathcal{R}}$.

Definition 3. *(Rewriting system based on a set of clauses) Let S be a set of clauses. The rewriting system $\mathcal{R}(S)$ based on S is a rewriting system on atoms defined by the set of rewriting rules $L \rightarrow R$ such that L and R are two atoms of C with $C \in S$ and $L \succeq_a R$.*

We remark that when S is finite $\mathcal{R}(S)$ is also finite, and $S \subseteq S'$ implies $\mathcal{R}(S) \subseteq \mathcal{R}(S')$.

We now deviate from the traditional notion of *refutational proof* and define instead the notion of *local dag proof*. Informally, a *refutational proof* of $S \cup \neg C$ for a set S of clauses and a clause C is a tree where leaves are labeled by ground instances of clauses in $\{S \cup \neg C\}$, internal nodes are labeled by the conclusion of the resolution applied to the antecedent nodes, and the root is labeled by the empty clause. In the *dag proof* we introduce an ordering on the nodes such that the leaves are minimal and the root is maximal with respect to this new ordering.

Definition 4. *(Dag proofs) Given a set S of clauses, a clause C and an ordered finite set of ground clauses $(T, <_T)$. We call $(T, <_T)$ a dag proof of $S \cup \neg C$ if:*

1. *for any clause $t \in T$, we have either t is a ground instance of a clause in $S \cup \neg C$, or there exists $t_1, t_2 \in T$ with $t_1, t_2 <_T t$ and t is the conclusion of the resolution applied to t_1 and t_2.*
2. *T contains the empty clause.*

When such $(T, <_T)$ exists, we write $S \vdash C$. In a dag proof, each minimal clause with respect to the ordering $<_T$ is called a leave.

Definition 5. *(Local dag proofs) Given a set S of clauses, a clause C, an ordered finite set of ground clauses $(T, <_T)$ and a set $\mathcal{A}$ of ground atoms. We call $(T, <_T)$ a $\mathcal{A}$-local dag proof of $S \cup \neg C$ if $(T, <_T)$ is a dag proof of $S \cup \neg C$ and $Atoms(T) \subseteq \mathcal{A}$. When such $(T, <_T)$ and $\mathcal{A}$ exist, we write $S \vdash_{\mathcal{A}} C$.*

Lemma 3. *Given a finite set S of clauses, a ground clause C and a finite rewriting system on atoms $\mathcal{R}$, we can decide whether $S \vdash_{C\downarrow_{\mathcal{R}}} C$.*

Proof. $\mathcal{R}$ is finite, and C is ground, this implies that $C\downarrow_{\mathcal{R}}$ is finite and ground (Lemma 2). For each $C\downarrow_{\mathcal{R}}$ local dag proof of $S \cup \neg C$, leaves are in a finite set of ground clauses, and the set of these leaves is unsatisfiable. The problem consisting is determining whether a finite set of ground clauses is unsatisfiable is decidable, and hence we can decide whether there exists a $C\downarrow_{\mathcal{R}}$ local dag proof of $S \cup \neg C$.

We define a notion of *redundancy* that identifies clauses and inferences that are not needed for performing the saturation procedure.

Definition 6. *(Redundancy) Let $\mathcal{R}$ be a finite rewriting system on atoms, a ground clause C is called $\mathcal{R}$-redundant in a set S of clauses if $S \vdash_{C\downarrow_{\mathcal{R}}} C$, a non-ground clause C is called $\mathcal{R}$-redundant in a set S of clauses if all its ground instances are $\mathcal{R}$-redundant in S, and an inference $C', C" \leadsto C$ by ordered resolution is called $\mathcal{R}$-redundant in the set S of clauses if (1) one of the premises (C' and $C"$) is $\mathcal{R}$-redundant in S, or else if (2) $S \vdash_{C\downarrow_{\mathcal{R}}} C$.*

Note that under this definition of redundancy, if a clause C in S is subsumed by a clause C' in S then C is $\mathcal{R}$-redundant in S.
Using this notion of redundancy, we show next how to relate *a priori* and *a posteriori* ordered resolution rules.

Lemma 4. *Let $C_1 = \Gamma_1 \rightarrow \Delta_1, A_1$ and $C_2 = A_2, \Gamma_2 \rightarrow \Delta_2$ be two clauses, $C_1, C_2 \leadsto C$ be an inference by priori ordered resolution with $A_1\sigma$ the resolved atom, and $\mathcal{R} = \mathcal{R}(C_1\sigma) \cup \mathcal{R}(C_2\sigma)$. Then either this inference is $\mathcal{R}$-redundant in $\{C_1, C_2\}$ or is an inference by posteriori ordered resolution.*

Proof. We have $C_1 = \Gamma_1 \rightarrow \Delta_1, A_1$, $C_2 = A_2, \Gamma_2 \rightarrow \Delta_2$, and $C_1, C_2 \leadsto C$ with $C = \Gamma_1\sigma, \Gamma_2\sigma \rightarrow \Delta_1\sigma, \Delta_2\sigma$ be an inference by *priori* ordered resolution. We assume that $C_1, C_2 \leadsto C$ is not an inference by *posteriori* ordered resolution. Then either $A_1\sigma$ is not strictly maximal for $\succ_a$ in the set of atoms of $C_1\sigma$, or $A_1\sigma$ is not maximal for $\succ_a$ in the set of atoms of $C_2\sigma$. This implies that there is an atom B in C with $A_1\sigma \preceq_a B$. Let j be such that $B \in Atoms(C_j\sigma)$. $C_j\sigma$ contains $A_1\sigma$ and B with $A_1\sigma \preceq_a B$. This implies that $B \rightarrow A_1\sigma \in \mathcal{R}(C_j\sigma)$, and hence $A_1\sigma\sigma_{S,C} \in Atoms(C\sigma_{S,C})\downarrow_{\mathcal{R}}$ with $S = \{C_1, C_2\}$. We then have $C_1, C_2 \vdash_{C\sigma_{S,C}\downarrow_{\mathcal{R}}} C$, and hence the inference $C_1, C_2 \leadsto C$ is $\mathcal{R}$-redundant in $\{C_1, C_2\}$.

5 Saturation

Definition 7. *(Saturated set of clauses) Let $\mathcal{R}$ be a rewriting system on atoms. We say that a set S of clauses is $\mathcal{R}$-saturated up to redundancy by ordered resolution if* (1) *any inference by priori ordered resolution from premises in S is $\mathcal{R}$-redundant in S,* (2) *$\mathcal{R}(S) \subseteq \mathcal{R}$, and* (3) *for each priori ordered resolution inference $C_1, C_2 \rightsquigarrow C$ with $C_1, C_2 \in S$, if the resolved atom $A\sigma$ is not strictly maximal in $C_1\sigma$ or not maximal in $C_2\sigma$ then $\mathcal{R}(\{C_1\sigma, C_2\sigma\}) \subseteq \mathcal{R}$.*

We present now a procedure that, providing it terminates, constructs from a finite set S of clauses a pair $(S', \mathcal{R})$ such that S' is a finite set of clauses, $\mathcal{R}$ is a rewriting system on atoms, and for every ground clause C, we have $S \models C$ iff $S' \vdash_{C\downarrow_{\mathcal{R}}} C$.

Input:
A finite set S of clauses.
Initialization:
Let $(S_1, \mathcal{R}_1) = (S, \mathcal{R}(S))$, and $i = 1$.
Transformation step:
We construct the pair $(S_{i+1}, \mathcal{R}_{i+1})$ from the pair $(S_i, \mathcal{R}_i)$ as follows: Let $C_1, C_2 \rightsquigarrow C$ be an inference by ordered resolution with $C_1, C_2 \in S_i$, and $A\sigma$ the resolved atom; One of the following three cases will be applied:

- *Non-maximality:* If $A\sigma$ is not strictly maximal for $\succ_a$ in the atoms of $C_1\sigma$ or not maximal for $\succ_a$ in the atoms of $C_2\sigma$ then $S_{i+1} = S_i$, $\mathcal{R}_{i+1} = \mathcal{R}_i \cup \mathcal{R}(\{C_1\sigma, C_2\sigma\})$, and $i = i + 1$;
- *Redundancy:* Otherwise, if $S_i \vdash_{C\downarrow_{\mathcal{R}_i}} C$ then $S_{i+1} = S_i$, $\mathcal{R}_{i+1} = \mathcal{R}_i$, and $i = i + 1$;
- *Discovery:* Otherwise a new clause useful for establishing local proofs has been discovered, and hence $S_{i+1} = S_i \cup \{C\}$, $\mathcal{R}_{i+1} = \mathcal{R}_i \cup \mathcal{R}(C)$, and $i = i + 1$.

Iteration:
We repeat the **Transformation step** until a fixed point is reached.

Returns $(S_i, \mathcal{R}_i)$.

Fig. 3: Saturation procedure

Definition 8. *The saturation procedure is called* fair *when every possible inference by priori ordered resolution has been performed.*

From now on, we only consider fair saturation procedure and we may omit the word "fair" for simplicity.
We prove next that the saturation procedure actually constructs a saturated set of clauses.

Proposition 1. *Let S be a finite set of clauses and $(S', \mathcal{R})$ be the output of the saturation procedure. S' is $\mathcal{R}$-saturated.*

Proof. Let S be a finite set of clauses such that the saturation procedure terminates and outputs $(S', \mathcal{R})$. By the initialization and discovery cases of the saturation, we have $\mathcal{R}(S') \subseteq \mathcal{R}$, and by the non-maximality case of the saturation procedure we have $\mathcal{R}(\{C_1\sigma, C_2\sigma\}) \subseteq \mathcal{R}$ for each $C_1, C_2 \in S'$ on which *priori* ordered resolution is possible but not *posteriori* ordered resolution. Now, we prove that any inference by ordered resolution from premises in S' is $\mathcal{R}$-redundant in S'. Let $C_1, C_2 \rightsquigarrow C$ be an inference by ordered resolution with $C_1, C_2 \in S'$. Since the saturation is fair, this inference has been considered during the computation of $(S', \mathcal{R})$, and falls into one of the following cases: the *redundancy*, the non-maximality, the *discovery*. By contradiction, assume that $C_1, C_2 \rightsquigarrow C$ is not $\mathcal{R}$-redundant in S', then we fall in one of the two other cases:

non-maximality: the resolved atom $A\sigma$ is not strictly maximal in the atoms of C. Therefore $C_1, C_2 \rightsquigarrow C$ is not an inference by *posteriori* ordered resolution, and by construction $\mathcal{R}(C_1\sigma) \cup \mathcal{R}(C_2\sigma) \subseteq \mathcal{R}$. Furthermore, Lemma 4 implies that the inference is $\mathcal{R}(C_1\sigma) \cup \mathcal{R}(C_2\sigma)$-redundant, and hence it is $\mathcal{R}$-redundant, which contradicts our assumption of non-redundancy.

discovery: this case implies that $C \in S'$, and then it is trivial that the inference is $\mathcal{R}$-redundant in S', which contradicts our assumption of non-redundancy.

As a consequence every inference between two clauses of S' must be $\mathcal{R}$-redundant. We finally conclude that S' is $\mathcal{R}$-saturated.

6 Decidability of the ground entailment problem

We consider in this section a finite set S of clauses, and a finite rewriting system $\mathcal{R}$ on atoms such that S is $\mathcal{R}$-saturated.

Proposition 2. *Let C be a ground clause. We have that $S \models C$ implies $S \vdash_{C\downarrow_{\mathcal{R}}} C$.*

Proof. Let $\mathcal{R}$ be a finite rewrite system on atoms, S be a finite set of clauses which is $\mathcal{R}$-saturated, and C be a ground clause such that $S \models C$. Let Π be a set of DAG proofs of $S \cup \neg C$. Since the resolution is complete and correct, we have $\Pi \neq \emptyset$. For every $\pi \in \Pi$, let $\delta(\pi) = Atoms(\pi) \downarrow_{\mathcal{R}} \setminus Atoms(C) \downarrow_{\mathcal{R}}$ be the distance from π to a local dag proof (if $\delta(\pi) = \emptyset$ then π is a local dag proof).

Let $\pi \in \Pi$ be such that $\delta(\pi)$ is minimal, and let us prove that $\delta(\pi) = \emptyset$. By contradiction, assume that $\delta(\pi) \neq \emptyset$ and let A be a maximal atom in $\delta(\pi)$ for the ordering $\prec_a$. By Lemma 1, we have that $B \to_{\mathcal{R}} B'$ implies that $B \succeq_a B'$ and hence A is an atom of π. We prove in the next *claim* that A must be maximal with respect to the atoms of π for the ordering $\prec_{\mathcal{R}}$.

Claim 1. The atom A is maximal in $Atoms(\pi)$ for the ordering $\prec_{\mathcal{R}}$.

Proof of the claim. *By contradiction if this were not the case there would exist an atom $B \in Atoms(\pi)$ with $B \neq A$, $A \prec_{\mathcal{R}} B$, and hence $A \prec_a B$ (Lemma 1). Since A is maximal in $\delta(\pi)$ for the ordering $\prec_a$, we would have that B is not in $Atoms(\pi)\downarrow_{\mathcal{R}} \setminus Atoms(C)\downarrow_{\mathcal{R}}$, and thus $B \in Atoms(C)\downarrow_{\mathcal{R}}$. Since $A \prec_a B$, we have that $A \in Atoms(C)\downarrow_{\mathcal{R}}$, which contradicts $A \in \delta(\pi)$.*

Let $Leaves^{+}_{A}$ be the set of leaves of π that contain the atom A, and $Leaves^{-}_{A}$ be the set of leaves that do not contain A. The correctness and completeness of the resolution implies that the set of clauses $Leaves^{+}_{A} \cup Leaves^{-}_{A}$ is unsatisfiable.

Claim 2. Each clause $C_A \in Leaves^{+}_{A}$ is an instance with a substitution σ of a clause $C^s_A \in S$ with every atom A^s satisfying $A^s\sigma = A$ is maximal for $\succ_a$.

Proof of the claim. *By definition of $Leaves^{+}_{A}$, C_A is either a ground instance of a clause in S or a clause in $\neg C$. Since A is not an atom occurring in C the later case is excluded. Thus there exists a clause $C^s_A \in S$, an atom $A^s \in C^s_A$, and a substitution σ such that $A^s\sigma = A$ and $C^s_A\sigma = C_A$. Finally if A^s is not maximal for $\succ_a$ in C^s_A and $\mathcal{R}(S) \subseteq \mathcal{R}$ then it is not maximal for $\prec_{\mathcal{R}}$ in C^s_A and thus by monotonicity, A is not maximal for $\prec_{\mathcal{R}}$ in the atoms of C_A. This contradicts the fact that A is maximal for $\prec_{\mathcal{R}}$ among the atoms occurring in π.*

Thus every resolution on A between two clauses C, C' in $Leaves^{+}_{A}$ is a ground instance with substitution σ of a *priori* ordered resolution between two clauses C^s, C'^s in S. In π, $Leaves^{+}_{A}$ are the unique leaves containing A; Furthermore, A is maximal in each clause of $Leaves^{+}_{A}$ for the ordering $\succ_a$ by *Claim 2*. This implies that we can first eliminate all the occurrences of the atom A by application of the priori ordered resolution on $Leaves^{+}_{A}$, and let $Leaves'$ be the obtained set of clauses after performing all possible resolutions on A in $Leaves^{+}_{A}$. The unsatisfiability of $Leaves^{+}_{A} \cup Leaves^{-}_{A}$ implies that unsatisfiability of $Leaves' \cup Leaves^{-}_{A}$. We prove next that we can construct a new DAG proof π' of $S \cup \neg C$ with $\delta(\pi') \prec^{set}_{a} \delta(\pi)$.

Let $C = \Gamma \to \Delta, A$, $C' = A, \Gamma' \to \Delta'$ be two clauses in $Leaves^{+}_{A}$, and let C'' be the result of the resolution on C and C'. By definition of $Leaves'$, we have $C'' \in Leaves'$. Let C^s and C'^s be two clauses on S such that: $C = C^s\sigma$, $C' = C'^s\sigma$, $A^s \in Atoms(C^s)$ with A^s maximal in C^s for $\succ_a$ and $A = A^s\sigma$, $A'^s \in Atoms(C'^s)$ with A'^s maximal in C'^s for $\succ_a$ and $A = A'^s\sigma$. Wlog, assume that ordered factorization has been applied to C^s and C'^s so that there is one-to-one mapping between atoms of C^s (respectively C'^s) and atoms of C (respectively atoms of C'). The *priori* ordered resolution can then be applied to C^s and C'^s with θ the most general unifier of A^s and A'^s, and C''^s is the obtained clause. Since S is $\mathcal{R}$-saturated, this inference is saturated and then one of the two following cases holds:

1. $C''^s \in S$: C'' is then a ground instance of a clause in S. In this case we let $S(C'') = \{C''\}$. We remark that $Atoms(C'') = Atoms(C \cup C') \setminus \{A\} \subseteq Atoms(\pi) \setminus \{A\}$.

2. $C''^s \notin S$: by the *saturation procedure*, we have two cases:
   (a) Non-maximality: the inference $C^s, C'^s \rightsquigarrow C''^s$ is not an inference by *posteriori* ordered resolution, and hence by Lemma 4, the inference is $\mathcal{R}(C^s\theta \cup C'^s\theta)$-redundant in $\{C^s, C'^s\}$, and then $\mathcal{R}$-redundant in $\{C^s, C'^s\}$. By definition of the *redundancy*, we then have $S \vdash_{C''\downarrow_\mathcal{R}} C''$. We then let $S(C'')$ be a set ground instances of clauses of S whose atoms are in $C'' \downarrow_\mathcal{R}$ that entails C''. We remark that $Atoms(S(C'')) \subseteq C'' \downarrow_\mathcal{R} \subseteq (Atoms(\pi) \setminus A) \downarrow_\mathcal{R}$.
   (b) Redundancy: C''^s is $\mathcal{R}$-redundant in S, and then by Definition 6, all ground instances of C''^s are $\mathcal{R}$-redundant in S. This implies that $S \vdash_{C''\downarrow_\mathcal{R}} C''$. We let $S(C'')$ be a set ground instances of clauses of S whose atoms are in $C'' \downarrow_\mathcal{R}$ that entails C''. We remark that $Atoms(S(C'')) \subseteq C'' \downarrow_\mathcal{R} \subseteq Atoms(\pi) \setminus A \downarrow_\mathcal{R}$.

The unsatisfiability of $Leaves^+_A \cup Leaves^-_A$ implies the unsatisfiability of $Leaves^-_A \cup \bigcup_{C'' \in Leaves'} S(C'')$, and hence there is a DAG proof π' of $Leaves^-_A \cup \bigcup_{C'' \in Leaves'} S(C'')$, which is also a DAG proof of $S \cup \neg C$. We prove next that $\delta(\pi') \prec^{set}_a \delta(\pi)$.

$$
\begin{aligned}
\delta(\pi') &= Atoms(\pi') \downarrow_\mathcal{R} \setminus Atoms(C) \downarrow_\mathcal{R} \\
&= [Atoms(Leaves^-_A) \cup \bigcup_{C'' \in Leaves'} Atoms(S(C''))] \downarrow_\mathcal{R} \setminus Atoms(C) \downarrow_\mathcal{R} \\
&\subseteq (Atoms(\pi) \setminus A \cup Atoms(Leaves')) \downarrow_\mathcal{R} \setminus Atoms(C) \downarrow_\mathcal{R} \\
&\subseteq (Atoms(\pi) \downarrow_\mathcal{R} \setminus A) \setminus Atoms(C) \downarrow_\mathcal{R} \text{ (maximality of } A \text{ in } Atoms(\pi)) \\
&= (Atoms(\pi) \downarrow_\mathcal{R} \setminus Atoms(C) \downarrow_\mathcal{R}) \setminus A \\
&= \delta(\pi) \setminus A.
\end{aligned}
$$

Since $A \in \delta(\pi)$ and is maximal, we then have $\delta(\pi') \prec^{set}_a \delta(\pi)$, and hence, there is a DAG proof π' of $S \cup \neg C$ with $\delta(\pi')$ strictly smaller than $\delta(\pi)$ and that contradicts the minimality of $\delta(\pi)$. We conclude that $\delta(\pi) = \emptyset$, and hence we have $S \vdash_{C\downarrow_\mathcal{R}} C$.

Proposition 3. *Let C be a ground clause. We have that $S \vdash_{C\downarrow_\mathcal{R}} C$ implies $S \models C$.*

Proof. Let C be a ground clause such that $S \vdash_{C\downarrow_\mathcal{R}} C$. This implies that there is a DAG proof of $S \cup \neg C$, and hence by correctness of the resolution, $S \cup \neg C$ is unsatisfiable, and hence $S \models C$.

Proposition 4. *Let $\mathcal{R}$ be a finite rewriting system on atoms, and S be an $\mathcal{R}$-saturated set of clauses. The ground entailment problem for S is decidable.*

Proof. Let C be an arbitrary ground clause, the *ground entailment problem* for S is decidable if and only if $S \models C$ is decidable. By the propositions 2 and 3, we

have that $S \models C$ if and only if $S \vdash_{C\downarrow_{\mathcal{R}}} C$. By Lemma 3, $S \vdash_{C\downarrow_{\mathcal{R}}} C$ is decidable. We conclude that $S \models C$ is decidable, and hence the *ground entailment problem* is decidable.

From the previous lemmas and propositions, we obviously deduce the following theorem which is the main result of the paper.

Theorem 1. *Let $\succ_a$ be a well-founded, monotone atom ordering such that $A \prec_a B$ implies $\mathrm{Var}(A) \subseteq \mathrm{Var}(B)$ for every atoms A and B. Let S be a set of clauses such that the saturation on S terminates using the atom ordering $\succ_a$. Then the ground entailment problem for S is decidable.*

7 Comparison with existing works

This paper is meant to be an extension of [4] to more general orderings and it relies on *a priori* instead of *a posteriori* ordered resolution used in [4]. Though various settings are considered in [4], a common trait is that the atom ordering $\prec_a$ and the term ordering $\prec_t$ satisfy the following:

- the term ordering $\prec_t$ is well-founded and total on ground terms;
- the atom ordering $\prec_a$ is compatible with the term ordering $\prec_t$, *i.e.* $A(s_1, \dots, s_m) \prec_a B(t_1, \dots, t_n)$ whenever for any $1 \leq j \leq m$ there exists $1 \leq i \leq n$ such that $s_j \prec_t t_i$;
- the atom ordering $\prec_a$ is monotone;
- every term t has only a finite number of smaller terms for $\prec_t$.

We prove below that such orderings also satisfy our criteria when the underlying term ordering is subterm (*i.e.* $u[t] \succ t$ for every terms u and t), which is the case for term orderings such as KBO, LPO, RPO, etc.

Proposition 5. *If there exists an infinite number of terms and if the term ordering $\prec_t$ is* subterm *then under the above conditions $A \prec_a B$ implies $\mathrm{Var}(A) \subseteq \mathrm{Var}(B)$.*

Proof. Assume there exists a term t such that there does not exist t' with $t \prec_t t'$. Since the ordering is total on ground terms for every term $t' \neq t$ we have $t' \prec_t t$. Since there exists an infinite number of ground terms this contradicts the assumption that every term has only a finite number of terms smaller than itself. Thus for every term t there exists a term t' with $t \prec_t t'$.

Now let A and B be two atoms, and assume $\mathrm{Var}(A) \not\subseteq \mathrm{Var}(B)$. Let σ be a substitution grounding B, *i.e.*, $B\sigma = b(s_1, \dots, s_m)$. Wlog assume that s_1 is maximal among the $s_1, \dots, s_m$ for the term ordering $\prec_t$. Let t be a term greater than s_1. Let us extend σ on $\mathrm{Var}(A) \setminus \mathrm{Var}(B)$ by a substitution τ mapping every $x \in \mathrm{Var}(A) \setminus \mathrm{Var}(B)$ to t. Since there is at least one occurrence of one such x in A, and since the ordering is subterm, there exists a term t' in $A\sigma\tau$ that contains t as a subterm. Since the ordering is subterm this implies $t \prec_t t'$. Since the ordering on ground atoms is compatible with the ordering on ground terms this implies $B\sigma \prec_a A\sigma\tau$. Thus $\mathrm{Var}(A) \not\subseteq \mathrm{Var}(B)$ implies $A \not\prec_a B$.

Finally the assumptions employed in [4] to derive complexity results imply that the number of atoms smaller than a given ground atom of size n is in $\mathcal{O}(f(n))$ and such atoms may be enumerated in time $\mathcal{O}(g(n))$ for two computable functions f and g. Since we do not assume the same finiteness property we cannot directly state complexity results. However we note that there is a lot of works on the complexity analysis of term rewriting systems. While these works aim at bounding the maximal length of a derivation, we believe that it could still be useful to provide theoretic upper bounds on the number of atoms smaller than the atoms in a fixed set C for the constructed ordering $\prec_{\mathcal{R}}$.

8 Conclusion

We have presented in this paper an extension of a classical result by Basin and Ganzinger [4]. The relaxation of the hypothesis on the ordering lead to a further extension for resolution modulo an equational theory [8,13,15]. We note that the redundancy notion introduced in [1] is based on an ordering of clauses as multisets of literals. A drawback of the saturation procedure presented in this paper is that clauses are seen as sets of literals; Thus we cannot apply as is their result of combination of saturation with subsumption. We plan to prove in future works that it is possible to add to our saturation procedure a backward subsumption rule while preserving the construction of the finite complexity atom ordering.

We believe the technique employed can be extended to add a reflectivity or transitivity axiom to an already saturated theory. Also, we thank Chris Lynch [10] for having pointed to us (by giving a counter-example) that the method cannot be extended *as is* to superposition. Finally we believe that a consequence of our proof is that saturated theories are complete for contextual deduction [5,12], which may help in the resolution of [6], though further work is needed to confirm this conjecture.

References

1. Leo Bachmair and Harald Ganzinger. On restrictions of ordered paramodulation with simplification. In Mark E. Stickel, editor, *CADE*, volume 449 of *Lecture Notes in Computer Science*, pages 427–441. Springer, 1990.
2. Leo Bachmair and Harald Ganzinger. Completion of first-order clauses with equality by strict superposition (extended abstract). In *CTRS*, volume 516 of *Lecture Notes in Computer Science*, pages 162–180, 1991.
3. Leo Bachmair, Harald Ganzinger, Christopher Lynch, and Wayne Snyder. Basic paramodulation. *Inf. Comput.*, 121(2):172–192, 1995.
4. David Basin and Harald Ganzinger. Automated complexity analysis based on ordered resolution. *J. ACM*, 48(1):70–109, 2001.
5. Francois Bronsard and Uday S. Reddy. Conditional rewriting in focus. In M. Okada, editor, *Proceedings of the Second International Workshop on Conditional and Typed Rewriting Systems*, volume 516 of *Lecture Notes in Computer Science*. Springer-Verlag, 1991.

6. Nachum Dershowitz and Ralf Treinen. Rta list of open problems, problem 37. `http://rtaloop.mancoosi.univ-paris-diderot.fr/problems/summary.html`, 1998.
7. Jieh Hsiang and Michaël Rusinowitch. Proving refutational completeness of theorem-proving strategies: The transfinite semantic tree method. *J. ACM*, 38(3):559–587, 1991.
8. Gérard Huet. *Constrained Resolution: A Complete Method for Higher Order Logic*. PhD thesis, Case Western Reserve University, 1972.
9. Christopher Lynch. Schematic saturation for decision and unification problems. In Franz Baader, editor, *CADE*, volume 2741 of *Lecture Notes in Computer Science*, pages 427–441. Springer, 2003.
10. Christopher Lynch. Personnal communication. Toulouse, december 2009, 2009.
11. David A. McAllester. Automatic recognition of tractability in inference relations. *J. ACM*, 40(2):284–303, 1993.
12. Robert Nieuwenhuis and Fernando Orejas. Clausal rewriting. In *CTRS*, pages 246–258, 1990.
13. Robert Nieuwenhuis and Albert Rubio. Ac-superposition with constraints: No ac-unifiers needed. In *CADE*, pages 545–559, 1994.
14. John Alan Robinson. A machine-oriented logic based on the resolution principle. *J. ACM*, 12(1):23–41, 1965.
15. Laurent Vigneron. Associative-commutative deduction with constraints. volume 814 of *Lecture Notes in Computer Science*, pages 530–544, 1994.